Amendments to the Specification

1. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating an expanded conversion matrix from the data structure,

condensing the expanded conversion matrix to a condensed generic format, said condensed conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the condensed conversion matrix.

2. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix; and

wherein the HDL is Verilog, the data structure is a user defined primitive (UDP), and the generic HDL register and the generic HDL input logic consist entirely of Verilog primitives.

3. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix; and

wherein the specific format comprises a technology-specific format used to create a library of elements from which the circuit element is received.

4. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.

wherein the conversion matrix comprises a plurality of entries representing every state of the circuit element and every corresponding next state of the circuit element.

5. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix; and

wherein generating the conversion matrix comprises:

performing a reachability analysis on the conversion matrix to classify whether each particular state of the circuit element defined by the conversion matrix is reachable.

6. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.

wherein generating the conversion matrix comprises:

identifying input signals and an output signal of the circuit element from the data structure;

evaluating state transitions for the circuit element by evaluating the data structure for a next output signal for each transition of the input signals and for each current output signal; and

populating entries of the conversion matrix for each state transition.

7. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein generating the conversion matrix comprises:

identifying input signals and an output signal of the circuit element from the data structure;

evaluating state transitions for the circuit element by evaluating the data structure for a next output signal for each transition of the input signals and for each current output signal; and

populating entries of the conversion matrix for each state transition; and

wherein states for individual signals comprise 0, 1, and X.

8. (previously presented) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein the conversion matrix is organized into a plurality of current states of the circuit element and corresponding next states of the circuit element, and wherein determining the generic HDL register and the plurality of generic HDL input logic comprises:

classifying each next state of the conversion matrix into one of a plurality of sets of states based on predefined criteria;

selecting either an edge sensitive HDL primitive or a level sensitive HDL primitive for the generic HDL register based on selected ones of the plurality of sets of states;

selecting a particular set of functions based on the generic HDL register selected, each function of the particular set of functions corresponding to an input to the generic HDL register; and

evaluating the particular set of functions to determine the plurality of generic HDL input logic.

9. (currently amended) The method of claim 8 wherein each next state is classified according to level (L) or edge (E) sensitive states, according to reachable or unreachable (UR) states, wherein output state transitions comprise ~~and according to~~ the six categories of output state transitions ~~(QQ+ =~~ included within the following parentheses (00, 01, 0X, 10, 11, 1X) such

that the plurality of sets of states comprises 24 sets, and wherein the 24 sets of states comprise the sets which are included within the following parentheses (L00, L01, L0X, LI0, L11, L1X, E00, E0l, E0X, E10, E11, E1X, LUR00, LUR01, LUR0X, LUR10, LUR11, LUR1X, EUR00, EUR01, EUR 0X, EUR10, EUR11, EUR1X).

10. (currently amended) The method of claim 9 wherein each current state comprises N current input values, wherein N is a variable and corresponds to a quantity of input values, and a current output value Q, wherein each current state is classified as reachable or unreachable, wherein each next state comprises N next input values and a next output value Q+, wherein individual value states comprise one of 0, 1, and X, and wherein classifying each next state comprises:

selecting a particular next state having a particular output value Q+;

identifying any current states corresponding to the particular next state in the conversion matrix to provide identified current states;

ignoring any of the identified current states that have a current input value of X to provide a set of remaining current states;

organizing the remaining current states by their respective output values Q;

classifying the particular next state as a level sensitive state (L) for one or more of the said six categories of output state transitions ~~(QQ+)~~ for any value of Q for which N of the remaining current states have a same value Q;

classifying the particular next state as an edge sensitive state (E) for one or more of the said six categories of output state transitions ~~(QQ+)~~ for any value of Q for which less than N of the remaining current states have a same value Q; and

classifying the particular next state as unreachable (UR) for one or more of the said six categories of output state transitions ~~(QQ+)~~ for any value of Q for which all of the remaining current states are classified as unreachable.

11. (original) The method of claim 9 wherein selecting either the edge sensitive HDL primitive or the level sensitive HDL primitive comprises:

selecting the level sensitive HDL primitive if sets E01 and El0 are empty.

12. (currently amended) The method of claim 9 wherein the particular set of functions for the edge sensitive HDL primitive comprises:

Fset= L01 with do not cares for states which are a union of L11, LUR01, and LUR11;

Frst = L10 with do not cares for states which are a union of L00, LUR10, and LUR00;

Fclen = a single common input identified in sets E01 and E10;

Ftmpa = selected next states from a union of E01, E11, and L11;

Ftmpb = Ftmpa with do not cares for states which are selected next states from a union of ~~EURO1~~ EUR01, EUR11, and LUR11;

Fd = Ftmpb with do not cares for states which are a union of Fset and Frst, wherein the single common input identified for Fclen is ignored, and wherein the selected next states are edge states having a same edge transition as the single common input identified for Fclen.

13. (currently amended) The method of claim 9 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = L01 with do not cares for the states of L11;

Frst = L10 with do not cares for the states of L00;

Fclen = 0; and

Fd = 0.

14. (currently amended) The method of claim 9 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = 0;

Frst = 0;

Fclen = (L01 with do not cares for the states of L11) union with (L10 with do not cares for the states of L00); and

Fd = L01 with do not cares for the states of L11.

15. (currently amended) The method of claim 9 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = a single input identified from the ~~connectivity~~conversion matrix for which there is no current state for which the output Q+ of the corresponding next state is ~~one~~ zero;

Frst = a single input identified from the ~~connectivity~~conversion matrix for which there is no current state for which the output Q+ of the corresponding next state is ~~zero~~ one;

Ftmp1 = L01 with do not cares for states which are a union of L11, LUR01, and LUR11;

Ftmp2 = L10 with do not cares for states which are a union of L00, LUR10, and LUR00;

Fclen = a union of Ftmp1 and Ftmp2 with do not cares for a union of Fset and Frst; and

Fd = Ftmp1 with do not cares for an inverted Fclen.

16. (previously presented) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic comprising combinational gates for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.

17. (currently amended) A machine readable medium having stored thereon machine executable instructions to implement a method comprising:

receiving a data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined in a hardware description language (HDL) using a specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix; and

wherein the HDL is Verilog, the data structure is a user defined primitive (UDP), and the generic HDL input logic consists entirely of Verilog primitives.

18. (currently amended) A machine readable medium having stored thereon machine executable instructions to implement a method comprising:

receiving a data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined in a hardware description language (HDL) using a specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix; and

wherein the specific format comprises a technology-specific format used to create a library of elements from which the circuit element is received.

19. (previously presented) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix wherein the conversion matrix comprises a plurality of entries representing every state of the circuit element and every corresponding next state of the circuit element.

20. (currently amended) A machine readable medium having stored thereon machine executable instructions to implement a method comprising:

receiving a data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined in a hardware description language (HDL) using a specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix; and

wherein generating the conversion matrix comprises performing a reachability analysis on the conversion matrix to classify whether each particular state of the circuit element defined by the conversion matrix is reachable.

21. (original) The machine readable medium of claim 16 wherein generating the conversion matrix comprises:

identifying input signals and an output signal of the circuit element from the data structure;

evaluating state transitions for the circuit element by evaluating the data structure for a next output signal for each transition of the input signals and for each current output signal; and

populating entries of the conversion matrix for each state transition.

22. (previously presented) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein generating the conversion matrix comprises:

identifying input signals and an output signal of the circuit element from the data structure;

evaluating state transitions for the circuit element by evaluating the data structure for a next output signal for each transition of the input signals and for each current output signal; and

populating entries of the conversion matrix for each state transition; and

wherein states for individual signals comprise 0, 1, and X.

23. (previously presented) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein the conversion matrix is organized into a plurality of current states of the circuit element and corresponding next states of the circuit element, and wherein determining the generic HDL register and the plurality of generic HDL input logic comprises:

classifying each next state of the conversion matrix into one of a plurality of sets of states based on predefined criteria;

selecting either an edge sensitive HDL primitive or a level sensitive HDL primitive for the generic HDL register based on selected ones of the plurality of sets of states;

selecting a particular set of functions based on the generic HDL register selected, each function of the particular set of functions corresponding to an input to the generic HDL register; and

evaluating the particular set of functions to determine the plurality of generic HDL input logic.

24. (currently amended) The machine readable medium of claim 23 wherein each next state is classified according to level (L) or edge (E) sensitive states, according to reachable or

unreachable (UR) states, wherein output state transitions comprise ~~and according to~~ the six categories of output state transitions ~~(QQ+=~~ included within the following parentheses (00, 01, 0X, 10, 11, 1X) such that the plurality of sets of states comprises 24 sets, and wherein the 24 sets of states comprise the sets which are included within the following parentheses (L00, L01, L0X, L10, L11, L1X, E00, E01, E0X, E10, E11, E1X, LUR00, LUR01, LUR0X, LUR10, LUR11, LUR1X, EUR00, EUR01, EUR 0X, EUR10, EUR11, EUR1X).

25. (currently amended) The machine readable medium of claim 24 wherein each current state comprises N current input values, wherein N is a variable and corresponds to a quantity of input values, and a current output value Q, wherein each current state is classified as reachable or unreachable, wherein each next state comprises N next input values and a next output value Q+, wherein individual value states comprise one of 0, I, and X, and wherein classifying each next state comprises:

selecting a particular next state having a particular output value Q+;

identifying any current states corresponding to the particular next state in the conversion matrix to provide identified current states;

ignoring any of the identified current states that have a current input value of X to provide a set of remaining current states;

organizing the remaining current states by their respective output values Q;

classifying the particular next state as a level sensitive state (L) for one or more of the said six categories of output state transitions ~~(QQ+)~~ for any value of Q for which N of the remaining current states have a same value Q;

classifying the particular next state as an edge sensitive state (E) for one or more of the said six categories of output state transitions ~~(QQ+)~~ for any value of Q for which less than N of the remaining current states have a same value Q; and

classifying the particular next state as unreachable (UR) for one or more of the said six categories of output state transitions ~~(QQ+)~~ for any value of Q for which all of the remaining current states are classified as unreachable.

26. (currently amended) The machine readable medium of claim 24 wherein selecting either the edge sensitive HDL primitive or the level sensitive HDL primitive comprises:

selecting the level sensitive HDL primitive if sets ~~E01~~ E01 and E10 are empty.

27. (currently amended) The machine readable medium of claim 24 wherein the particular set of functions for the edge sensitive HDL primitive comprises:

Fset = L01 with do not cares for states which are a union of L11, LUR01, and LUR11;

Frst = L10 with do not cares for states which are a union of L00, LUR10, and LUR00;

Fclen = a single common input identified in sets E01 and E10;

Ftmpa = selected next states from a union of E01, E11, and L11;

Ftmpb = Ftmpa with do not cares for states which are selected next states from a union of EUR01, EUR11, and LUR11;

Fd = Ftmpb with do not cares for states which are a union of Fset and Frst, wherein the single common input identified for Fclen is ignored, and wherein the selected next states are edge states having a same edge transition as the single common input identified for Fclen.

28. (currently amended) The machine readable medium of claim 24 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = L01 with do not cares for the states of L11;

Frst = L10 with do not cares for the states of L00;

Fclen = 0; and

Fd = 0.

29. (currently amended) The machine readable medium of claim 24 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = 0;

Frst = 0;

Fclen = (L01 with do not cares for the states of L11) union with (L10 with do not cares for the states of L00); and

Fd = L01 with do not cares for the states of L11.

30. (currently amended) The machine readable medium of claim 16 wherein the particular set of functions for the level sensitive hardware description language (HDL) primitive comprises:

Fset = a single input identified from the ~~connectivity~~conversion matrix for which there is no current state for which the output Q+ of the corresponding next state is ~~one~~ zero;

Frst = a single input identified from the ~~connectivity~~conversion matrix for which there is no current state for which the output Q+ of the corresponding next state is ~~zero~~ one;

Ftmp1 = L01 with do not cares for states which are a union of L11, LUR01, and LUR11;

Ftmp2 = L10 with do not cares for states which are a union of L00, LUR10, and LUR00;

Fclen = a union of Ftmp1 and Ftmp2 with do not cares for states which are a union of Fset and Frst; and

Fd = Ftmp1 with do not cares for states which are an inverted Fclen.

31. (previously presented) An apparatus for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the apparatus comprising:

a processor; and

a machine readable storage medium storing thereon machine executable instructions, the processor to execute the machine executable instructions to

receive the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generate a conversion matrix from the data structure, said conversion matrix to represent the data structure in a first compressed format;

convert the conversion matrix into a second condensed format, said condensed conversion matrix to represent the behavior of the circuit element in a generic format; and

determine a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.

32. (previously presented) The method of claim 1 whereby the condensing further comprises using a Karnaugh map.

33. (previously presented) The machine readable medium of claim 16, further comprising evaluating the conversion matrix, whereby evaluating comprises classifying a conversion matrix column into a category, the category comprising an edge state, a level state, a reachable state, or an unreachable state.

34. (previously presented) The machine readable medium of claim 33, where evaluating further comprises classifying a column as an output state transition.

35. (currently amended) The machine readable medium of claim 34, ~~where~~wherein output state transitions are represented by QQ+ and wherein the output state transition QQ+ comprises one of QQ+ = 00, QQ+=01, QQ+= 0X, QQ+= 10, QQ+= 11, or QQ+= 1X.

36. (currently amended) The method of claim 1 wherein determining provides identical generic HDL registers for data structures with different representations but identical functionality.

37. (previously presented) The method of claim 8 wherein each next state is classified as level (L) or edge (E) sensitive.

38. (previously presented) The method of claim 8, wherein each next state is classified as reachable (R) or unreachable (U).

39. (currently amended) The method of claim 8, wherein output state transitions comprise the output state transitions set forth within the following parentheses (00, 01, 0X, 10, 11, 1X, X0, X1, XX) and wherein each next state is classified using at least six of the ~~following~~ output state transitions set forth within the following parentheses ~~(QQ+ =~~ (00, 01, 0X, 10, 11, 1X, X0, X1, XX).

40. (currently amended) The method of claim 8, wherein output state transitions comprise the output state transitions set forth within the following parentheses (00, 01, 0X, 10, 11, 1X, X0,

X1, XX) and wherein each next state classification does not include at least one of the ~~following~~ output state transitions set forth in the following parentheses ~~(QQ+ =~~ (X0, X1, XX).

41. (currently amended) The apparatus of claim 31, wherein ~~data structures with different representations but identical functionality in~~ the data structure comprises first and second user defined primatives (UDPs) in a first condensed format which have different representations in the first condensed format but identical functionality, the conversion matrix representing the first and second user defined primitives in the first compressed format ~~are,~~ and wherein the first and second primitives represented by the conversion matrix are given identical representations in the second ~~compressed~~ condensed format.

42. (previously presented) The machine readable medium of claim 16, wherein the conversion matrix comprises a plurality of entries representing every state of the circuit element.

43. (currently amended) The machine readable medium of claim 16, wherein the circuit element has reachable states and wherein the plurality of entries representing every state of the circuit element further comprise every next reachable state of the circuit element.